Modeling Large S-box in MILP and a (Related-key) Differential Attack on Full Round PIPO-64/128

Tarun Yadav and Manoj Kumar

Scientific Analysis Group, DRDO, Metcalfe House Complex, Delhi-110 054, INDIA
{tarunyadav,manojkumar}@sag.drdo.in

Abstract. Mixed integer linear programming (MILP) based tools are used to estimate the strength of block ciphers against the cryptanalytic attacks. The existing tools use partial difference distribution table (p-DDT) approach to optimize the probability of differential characteristics for large ($\geq$ 8-bit) S-box based ciphers. We propose to use the full difference distribution table (DDT) with the probability of each possible propagation for MILP modeling of large S-boxes. This requires more than 16 variables to represent the linear inequalities of each propagation and corresponding probabilities. The existing tools (viz. Logic Friday) cannot handle the linear inequalities in more than 16 variables. In this paper, we present a new tool (namely MILES) to minimize the linear inequalities in more than 16 variables. This tool reduces the number of inequalities by minimizing the truth table corresponding to the DDT of S-box. We use our tool to minimize the linear inequalities for 8-bit S-boxes (AES and SKINNY) and get better results than existing tools. We show the application of MILES on 8-bit S-box based lightweight block cipher PIPO. There are 20621 inequalities in 23 variables corresponding to the possible propagations in DDT and these are minimized to 6035 inequalities using MILES. MILP model based on these linear inequalities is used to optimize the probability of differential characteristics for round-reduced PIPO. MILP model based on these inequalities is used to optimize the probability of differential and impossible differential characteristics for PIPO-64/128 reduced to 9 and 4 rounds respectively. We present an iterative 2-round related-key differential characteristic with the probability of 2^{-4} and that is used to construct a full round related-key differential distinguisher with the probability of 2^{-24}. We present a major collision in PIPO-64/128 which produces the same ciphertext (C) by encrypting the plaintext (P) under two different keys.

Keywords: Block Cipher, Differential Cryptanalysis, MILP, S-box

1 Introduction

Differential attack is one of the most powerful techniques for cryptanalysis of block ciphers [6]. For new block ciphers, it is a mandatory design criterion to provide the proof of resistance against the differential attack [5]. The high probability relations between the input and output differences of a block cipher are utilised to distinguish it from the uniform distribution [7]. We need a differential characteristic with probability of 2^{-p}, where $p \lll n$, to mount the attack on n-bit block cipher [14]. To estimate the strength of a block cipher against differential attack, we calculate a lower bound on the number of active S-boxes in a differential characteristic. Then, upper bound on the probability is estimated using this lower bound and maximum differential probability of the S-box [16]. Initially, branch-and-bound based techniques were used to search the high probability differential characteristics [20] [17]. Nowadays, automated solvers based on mixed integer linear programming (MILP) [21], satisfiability modulo theory (SAT/SMT) [8], constraint programming (CP) problems [11] [29], and machine learning based techniques [12] [30] are used to test the differential attack resistance. In 2012, MILP aided differential cryptanalysis for block ciphers was proposed by Mouha et al. This technique proved to be very successful to mount the differential attack on block ciphers.

Mixed integer linear programming is used frequently to solve the optimization problems. MILP deals with optimizing the objective function $f(x_1, x_2, \cdots, x_n)$ subject to a set of linear inequalities $Ax \leq b$ which involves decision variables $x_i, 1 \leq i \leq n$ with restrictions on certain variables to take integer values. We can convert the differential characteristic search problem into an MILP model [21]. Then, optimization problem solvers (viz. Gurobi [13] and CIPLEX [9]) are used to solve the MILP model to get a lower bound on the number of active S-boxes and search for high probability differential characteristics. The linear layers (viz. key addition and permutation layer) of a block cipher are easily converted into the linear inequalities. S-box is the non-linear component of a block cipher and DDT of the S-box is used to write the linear inequalities for each possible propagation. This set contains a large number of inequalities and it becomes hard to solve the MILP model based on this set. Therefore it is required to minimize the number of inequalities to obtain the solution efficiently. Various methods have been proposed in the literature to optimize the number of inequalities in this set.

Mouha et al. showed the use of MILP in differential cryptanalysis of block ciphers and used optimization solvers to get the security bounds [21]. They presented a framework to get the least number of active S-boxes in a differential characteristic of word oriented ciphers. This technique was illustrated on Advanced Encryption Standard (AES) and least number of active S-boxes in 4-round differential characteristic of AES were obtained by solving the MILP model.

Sun et al. extended the use of MILP for bit oriented ciphers and two methods based on logical condition modeling and convex hull computation were proposed to get the MILP model of S-box [23] [24]. The DDT of S-box was used to write the linear inequalities for possible propagations using the SageMath tool [25]. Then greedy search algorithm was used to reduce the number of inequalities in this model. For a 4-bit S-box, the reduced set contains about 30 inequalities. Due to limitation of SageMath, this method is not practical for the S-box of size greater than 6-bit.

Sasaki and Todo proposed another method for MILP modeling of S-box to reduce the number of inequalities [26] [27] [28]. They proposed MILP based method to reduce the inequalities using impossible propagations in the DDT of the S-box. For a 4-bit S-box, this method provides around 20 linear inequalities which is used in the first phase of MILP search. This method also uses SageMath to write the inequalities, therefore it also does not work for S-boxes of size more than 6-bit.

Abdelkhalek et al. proposed a method to generate a bit-wise model of the DDT for large (8-bit) S-boxes [1]. They utilized the relations between logical condition model and product-of-sum representation of Boolean functions and generated the constraint inequalities using Logic Friday [19] (which uses the Espresso algorithm). For 8-bit S-box, 16 variables are needed to represent the linear inequalities for possible and impossible propagation in the DDT during first phase search. This model was extended to search for high differential characteristic by separating the DDT into multiple tables (p-DDTs) for each probability. The behavior of tables was controlled by adding the conditional constraints. In second phase search, p-DDT approach is used due to limitation of Logic Friday to handle the inequalities in more than 16 variables. This tool can reduce the number of inequalities in first phase but it is unable to handle the inequalities in more than 16 input variables for second phase search.

Our Contribution: Minimization of linear inequalities in more than 16 variables is a challenging part of MILP based differential characteristics search problem. The existing approach uses partial DDT (p-DDT) of S-box to generate the linear inequalities due to limitation of logic minimization tool Logic Friday. For large S-boxes, we need to handle linear inequalities in more than 16 variables. To overcome this challenge, we develop a tool namely MILES (MInimized Linear inEqualities for large S-box) which can minimize the linear inequalities in more than 16 variables. MILES is based on Espresso algorithm [10] which is used for logic minimization. We convert the DDT of S-box into a truth table and that is minimized using MILES. We use MILES to minimize the number of inequalities for 8-bit S-boxes and compare the results with p-DDT approach for AES and SKINNY. We show the application of MILES to search the high probability differential characteristics of lightweight block cipher PIPO [15]. Up to the authors' knowledge, there does not exist any paper in the literature that uses full difference distribution table to optimize the probability of differential characteristics for large ($\geq$8-bit) S-boxes.

Organisation: The paper is organised as follows. In Section 2, we discuss a method for MILP modeling of block ciphers with 8-bit S-boxes. We present a new tool (MILES) to minimize the number of linear inequalities and compare the results for AES and SKINNY S-boxes. In Section 3, we show the application of MILES to model the MILP problem to optimize the probability of differential characteristics in lightweight block cipher PIPO. The paper is summarised with conclusion in Section 4.

2 MILP Based Differential Characteristic Search

To search the differential characteristics of a block cipher, the problem of optimizing the probability of differential characteristics is converted into the MILP problem. The objective function is the optimization of probabilities subject to the constraints based on linear inequalities. SPN and Feistel based block ciphers mainly consist of round key addition, substitution and permutation layers. The key addition layer does not contribute in the MILP model to search the differential characteristics. The input and output variables corresponding to the permutation layer are easily represented by linear inequalities. The substitution layer uses a non-linear S-box which cannot be easily represented by linear inequalities. SageMath is a popular tool to obtain the linear inequalities using input and output difference points in the DDT. For an efficient MILP model, it is required to reduce the number of linear inequalities. Sasaki and Todo [27] [28] suggested impossible points based approach to design an MILP problem for the minimization of these inequalities. This approach was later used by many researchers to design the MILP models of various 4-bit S-boxes [31]. The linear inequalities of permutation and substitution layers are used to model the MILP problem, which is solved by MILP solver GUROBI [13] or CPLEX [9].

MILP based differential characteristics search is two stage process. Firstly, number of active S-boxes is minimized and then probability of differential characteristic is optimized using these active S-boxes. The outer and inner modules of MILP are designed corresponding to these stages. The outer module minimizes the number of active S-boxes while inner module optimizes the probabilities of differential characteristics. The method of using extra variable for each unique probability is used in the inner module [24]. This method simplifies MILP model at the cost of extra variables.

2.1 Modeling Large S-box

An S-box is a non-linear component and it is converted into linear inequalities to model the MILP problem. SageMath is used to generate the linear inequalities for DDT of the S-box. For m-bit S-box, the size of DDT is $2^m \times 2^m$ and it represents the number of occurrences of possible output differences corresponding to each input difference. SageMath uses the H-representation of convex hull to generate linear inequalities for the S-box. This methods works well for small S-boxes only. SageMath has limitation on the dimension ($\leq$12) of such convex hulls. Due to this limitation, large S-boxes cannot be converted into the linear inequalities using SageMath. This limits the outer module of MILP model upto 6-bit S-boxes and inner module of MILP model upto 4-bit S-boxes.

For large (8-bit) S-boxes, Abdelkhalek et al. [1] addressed this problem using the Logic Friday [19] that reduces the inequalities by minimizing the product of sum of boolean functions. This technique was used to minimize the number of active S-boxes for 8-bit S-box based ciphers. Logic Friday has limitation on the number ($\leq$16) of input variables. Due to this limitation, Logic Friday can only be used to design the outer module but it cannot be used to design inner module for 8-bit S-box. Therefore, this method is used to find the number of active S-boxes only.

To optimize the probability of differential characteristic, Abdelkhalek et al. used p-DDT based approach by separating the DDT for each probability into multiple tables [1]. These tables can be represented with 16 input variables. Although, this method is used to optimize the probability of differential characteristic, it may not be efficient due to the of use of p-DDT instead of full DDT. Use of full DDT can reduce the number of linear inequalities which can be used to design an efficient MILP model to optimize the probability of differential characteristics. Based on the full DDT, Sun et al. [24] suggested the method of using extra variable for each unique probability. This method add extra input variables and there does not exist any tool which can handle more than 16 input variables for minimization of the linear inequalities. In this paper, we present a new tool namely MILES based on Espresso algorithm [10] and it can handle more than 16 variables to minimizes the set of linear inequalities. Linear inequalities generated from MILES are used to design the outer and inner modules of MILP model which optimizes the probability of differential characteristic.

2.2 MILES: MInimized Linear inEqualities for large S-boxes

We present a new tool MILES to generate the linear inequalities for larges S-boxes and this tool is based on the Espresso algorithm [10]. The S-box is given as an input to the tool and it outputs a minimized set of linear inequalities that is required to model the MILP problem. MILES is the first tool that uses the full DDT of 8-bit S-box to generate the linear inequalities. In MILES, there are four processes which are applied sequentially to generate the minimized linear inequalities. These process are described as follows:

1. **DDT Generation** - In this process, MILES takes m-bit S-box (m$\geq$3) as input and generates a DDT of the S-box. The DDT ($2^m \times 2^m$) is 2-Dimensional array where row indices(y-axis) define input difference while column indices(x-axis) define the output difference. We define a function $f_{i,j}$ to represent the DDT of S-box which provides the number of occurrences of output difference Δ_j corresponding to input difference Δ_i (equation 1).

$$f_{i,j} = Frequency_{\Delta_i \rightarrow \Delta_j} \text{where } 0 \leq i, j \leq m \tag{1}$$

   The process to generate the DDT from S-box is described in Algorithm 1. This DDT is used as an input in the next process.
2. **DDT to Truth Table conversion** - In this process, the input DDT is converted into a truth table. This truth table specifies the input and output points of the DDT as input variables. To simplify it, we specify only non-zero entries of the DDT and corresponding output variable as 1. MILES can generate three kinds of truth tables ($\star$-TT,p-TT,f-TT) from the DDT. The $*$-TT table corresponds to the non-zero entries in the DDT and p-TT corresponds to the non-zero entries in DDT for a specific probability (p). The f-TT table corresponds to the non-zero entries with extra input variable for

each probability. We describe the process to convert DDT into $*$-TT, p-TT and f-TT in Algorithm 2, 3 and 4 respectively.

3. **Truth Table minimization -** MILES interfaces with Espresso to perform minimization of the truth table. The output of minimization is TT_{min} which is used to generate the minimized linear inequalities. The TT_{min} is similar to the truth table and it contains an additional symbol ('$-$'). The output variable in TT_{min} is independent of input variable corresponding to this additional symbol. The minimization process can be performed with various modes available in Espresso algorithm. These options are chosen in MILES as minimization strategy. These strategies are problem specific and a particular strategy may not provide best solution for all problems. The minimized truth tables corresponding to $\star$-TT, p-TT, and f-TT are represented as $\star$-TT_{min}, p-TT_{min}, and f-TT_{min} respectively.
4. **Linear Inequalities Generation -** After minimization process, MILES generate the linear inequalities. Each linear inequality corresponds to one entry in TT_{min}. If a value in the entry is 0 then it is expressed as variable x and if it is 1 then it is expressed as $1 - x$. The value '$-$' in the TT_{min} does not contribute in the inequality generation process. The process to generate the minimized linear inequalities is described in the Algorithm 5.

Algorithm 1: DDT Generation

Input: $S = m$-bit S-box
Output: $\text{DDT}_{M\times M}$ $(M = 2^m)$
for i $\leftarrow$ 0 to $M-1$ **do**
  for j $\leftarrow$ 0 to $M-1$ **do**
    $in = i \oplus j$
    $out = S[i] \oplus S[j]$
    $\text{DDT}[in][out] \leftarrow \text{DDT}[in][out] + 1$
  end
end

Algorithm 2: DDT to $*$-TT Conversion

Input: $\text{DDT}_{M\times M}$
Output: $*$-TT
for i $\leftarrow$ 0 to $M-1$ **do**
  for j $\leftarrow$ 0 to $M-1$ **do**
    if $\text{DDT}[i][j] \neq 0$ **then**
      $*\text{-TT}.Add^a(concat^b(binary^c(i), binary(j))$
    end
  end
end

[a] A.Add(B): Adds an entry B to the truth table A
[b] concat(A,B(,C)): Returns an entry by concatenation of A, B and C(optional)
[c] binary(A): Returns binary representation of A

Algorithm 3: DDT to p-TT Conversion

```
Input: DDT_{M×M}, p
Output: p-TT
for i ← 0 to M − 1 do
    for j ← 0 to M − 1 do
        if DDT[i][j] = p then
            p-TT.Add(concat(binary(i), binary(j)))
        end
    end
end
```

Algorithm 4: DDT to f-TT Conversion

```
Input: DDT_{M×M}
Output: f-TT
N ← number of unique non-zero entries in f-DDT
V ← set of unique non-zero entries in f-DDT
V = v_z where z ∈ (1, N)
for i ← 0 to M − 1 do
    for j ← 0 to M − 1 do
        if DDT[i][j] = v_z then
            l ← Null
            for k ← 1 to N do
                if k = z then
                    concat(l, 1)
                end
                else
                    concat(l, 0)
                end
            end
            f-TT.Add(concat(binary(i), binary(j), binary(l)))
        end
    end
end
```

Algorithm 5: Linear Inequalities Generation

```
Input: TT_min
Output: L = set of linear inequalities
L ← Null
foreach entry in TT_min do
    LinIneq ← Null
    foreach variable(v_i) in the entry do
        if v_i = 0 then
            concat(LinIneq, x_i)
        end
        else
            concat(LinIneq,1 − x_i)
        end
        if v_i is the last variable then
            concat(LinIneq,−1,'≥ 0')
        end
        else
            concat(LinIneq, '+')
        end
    end
    L.Add(LinIneq)
end
```

2.3 Example: Linear Inequalities Generation using MILES

We describe the process to generate the linear inequalities for a 3-bit S-box (Table 1). The DDT (Table 2), f-TT (Table 3), and f-TT$_{min}$ (Table 4) are generated using MILES. The set of minimized linear inequalities for this S-box is given in Table 5.

x	0	1	2	3	4	5	6	7
S(x)	3	6	5	7	0	2	4	1

Table 1: 3-bit S-Box

	0	1	2	3	4	5	6	7
0	8	0	0	0	0	0	0	0
1	0	0	4	0	0	4	0	0
2	0	2	0	2	2	0	2	0
3	0	2	0	2	2	0	2	0
4	0	2	0	2	2	0	2	0
5	0	2	0	2	2	0	2	0
6	0	0	0	0	0	4	0	4
7	0	0	4	0	0	0	0	4

Table 2: DDT of S-Box

x_1	x_2	x_3	y_1	y_2	y_3	p_1	p_2	f
0	0	0	0	0	0	0	0	1
0	0	1	0	1	0	1	0	1
0	0	1	1	0	1	1	0	1
0	1	0	0	0	1	0	1	1
0	1	0	0	1	1	0	1	1
0	1	0	1	0	0	0	1	1
0	1	0	1	1	0	0	1	1
0	1	1	0	0	1	0	1	1
0	1	1	0	1	1	0	1	1
0	1	1	1	0	0	0	1	1
0	1	1	1	1	0	0	1	1
1	0	0	0	0	1	0	1	1
1	0	0	0	1	1	0	1	1
1	0	0	1	0	0	0	1	1
1	0	0	1	1	0	0	1	1
1	0	1	0	0	1	0	1	1
1	0	1	0	1	1	0	1	1
1	0	1	1	0	0	0	1	1
1	0	1	1	1	0	0	1	1
1	1	0	1	0	1	1	0	1
1	1	0	1	1	1	1	0	1
1	1	1	0	1	0	1	0	1
1	1	1	1	1	1	1	0	1

Table 3: f-TT of DDT

x_1	x_2	x_3	y_1	y_2	y_3	p_1	p_2	f
0	0	1	0	1	0	1	0	1
0	0	1	1	0	1	1	0	1
1	1	1	0	1	0	1	0	1
1	1	0	1	-	1	1	0	1
1	1	-	1	1	1	1	0	1
0	0	0	0	0	0	0	0	1
1	0	-	1	-	0	0	1	1
0	1	-	1	-	0	0	1	1
1	0	-	0	-	1	0	1	1
0	1	-	0	-	1	0	1	1

Table 4: f-TT$_{min}$ for f-TT

1	$x_1 + x_2 - x_3 + y_1 - y_2 + y_3 - p_1 + p_2 + 2 \geq 0$
2	$x_1 + x_2 - x_3 - y_1 + y_2 - y_3 - p_1 + p_2 + 3 \geq 0$
3	$-x_1 - x_2 - x_3 + y_1 - y_2 + y_3 - p_1 + p_2 + 4 \geq 0$
4	$-x_1 - x_2 + x_3 - y_1 - y_3 - p_1 + p_2 + 4 \geq 0$
5	$-x_1 - x_2 - y_1 - y_2 - y_3 - p_1 + p_2 + 5 \geq 0$
6	$x_1 + x_2 + x_3 + y_1 + y_2 + y_3 + p_1 + p_2 - 1 \geq 0$
7	$-x_1 + x_2 - y_1 + y_3 + p_1 - p_2 + 2 \geq 0$
8	$x_1 - x_2 - y_1 + y_3 + p_1 - p_2 + 2 \geq 0$
9	$-x_1 + x_2 + y_1 - y_3 + p_1 - p_2 + 2 \geq 0$
10	$x_1 - x_2 + y_1 - y_3 + p_1 - p_2 + 2 \geq 0$

Table 5: Linear inequalities generated from f-TT$_{min}$

2.4 Results for AES, Skinny, and PIPO S-boxes

We apply our tool MILES[1] on three 8-bit S-boxes used in block ciphers AES, SKINNY, and PIPO. We compare the number of linear inequalities generated by MILES with existing results based on $\star$-DDT and p-DDT approach. The $\star$-DDT is used to model MILP problem to minimize the number of active S-boxes. The $\star$-TT of MILES corresponds to $\star$-DDT and results for minimized linear inequalities are compared in Table 6. MILES gives better results by utilizing the various modes of Espresso Algorithm.

The approach based on p-DDT is used to design the linear inequalities of 8-bit S-box to optimize the probability of differential characteristics. In this approach, the DDT is separated into multiple p-DDT tables corresponding to different probabilities. Linear inequalities are generated for each p-DDT and all these inequalities are used to search the differential characteristics. The p-TT of MILES corresponds to p-DDT and results for minimized linear inequalities are compared in Table 7. It is evident from the results that MILES provides better reduction in the inequalities than existing tools.

MILES uses full DDT to minimize the number of linear inequalities for large S-boxes. The f-TT of MILES is used to reduce the inequalities and the results are compared in Table 8. For comparison with existing approach, we derive the number of linear inequalities of AES and SKINNY by adding linear inequalities of each p-TT. From the Table 8, it is apparent that MILES performs better than p-TT by utilizing full DDT at a time. The specific mode of MILES is also specified in the Table 8. Selection of a particular mode in MILES is not based on characteristic of S-box and each mode must be tried for better results.

Structure	# Inequalities QM ([1])	# Inequalities Espresso ([1])	# Inequalities MILES (This Paper)
AES S-box	-	8302	7899
SKINNY S-box	372	376	372
PIPO S-box	-	-	4474

Table 6: Number of linear inequalities to represent $\star$-DDT/$\star$-TT

3 Application to Lightweight Block Cipher PIPO

Lightweight cryptography has become an important topic in cryptology [2] and NIST has also called for a competition to design the lightweight cryptographic primitives [22]. PIPO is a lightweight block cipher which was recently proposed by Kim et al. at ICISC 2020 [15]. The design highlights are its security for side-channel protected and unprotected environments. Its diffusion layer is designed to

[1] https://github.com/tarunyadav/MILES

Structure (S-box)	Probability	QM ([1]) (p-DDT)	Espresso ([1]) (p-DDT)	MILES (This Paper) (p-TT)	MILES (Σp-TT)
AES	2^{-7}	-	8241	7927	8182
	2^{-6}	-	350	255	
SKINNY	2^{-7}	206	208	206	1123
	2^{-6}	275	283	275	
	$2^{-5.415037}$	33	34	33	
	2^{-5}	234	239	234	
	$2^{-4.415037}$	42	52	42	
	2^{-4}	147	159	147	
	$2^{-3.678071}$	15	15	15	
	$2^{-3.415037}$	24	28	24	
	$2^{-3.192645}$	15	15	15	
	2^{-3}	62	67	62	
	$2^{-2.678071}$	16	16	16	
	$2^{-2.415037}$	17	17	17	
	2^{-2}	37	40	37	
PIPO	2^{-7}	-	-	3410	6634
	2^{-6}	-	-	2211	
	$2^{-5.415037}$	-	-	519	
	2^{-5}	-	-	355	
	$2^{-4.678072}$	-	-	26	
	$2^{-4.415037}$	-	-	20	
	2^{-4}	-	-	93	

Table 7: Number of Linear Inequalities to represent p-DDT/p-TT

Structure (S-box)	MILES (mode)	# Inequalities ($\sum p$-TT)	# Inequalities (f-TT)
AES	strong	8182	8165
SKINNY	pos	1123	526
PIPO	pos	6634	6035

Table 8: Minimzed Linear inequalities using MILES

optimize the efficiency in hardware as well as software applications. Its diffusion layer can be implemented in software using the cyclic shift rotations. For hardware applications, its diffusion layer can be visualised as bit permutation on 64 bits and can be implemented using wirings only. The 8-bit S-box is specifically designed for PIPO so that it can be represented using the minimum number of non-linear equations. This also ensures the protection of the design against side channel attacks.

3.1 Specification of PIPO

PIPO is a 64-bit lightweight block cipher with 128 and 256 bits key sizes [15]. It consists of 13/17 rounds for 128/256 bits key variants respectively. It is based on substitution permutation network (SPN) structure. The lightweight 8-bit S-box, having differential branch number 3, is specifically designed to use in the confusion layer of PIPO. For each 8-bit word, diffusion layer uses a cyclic rotation with different shift values for each word. The round function of PIPO is explained by dividing it into an 8×8 matrix. It applies the diffusion layer row-wise and 8-bit S-box is applied column-wise. For each variant, a simple key selection algorithm is used. For 128-bit key $K = (K_1 \parallel K_0)$, the rounds keys are selected as $RK_i = K_{i(mod2)}, 0 \leq i \leq 13$. For 256-bit key $K = (K_3 \parallel K_2 \parallel K_1 \parallel K_0)$, the rounds keys are selected as $RK_i = K_{i(mod4)}, 0 \leq i \leq 17$.

Algorithm 6: Encryption Algorithm of PIPO

`Input:` P `and` $RK_i, 0 \leq i \leq 13$
`Output:` $C = (c_{63}, c_{62}, \cdots, c_0)$
$U_0 \leftarrow P \oplus RK_0$
$U_0 = (u_{63}, u_{62}, \cdots, u_0)$
for i=1 to 13 **do**
  for j=0 to 7 **do**
    $(v_{56+j} \parallel v_{48+j} \parallel v_{40+j} \parallel v_{32+j} \parallel v_{24+j} \parallel v_{16+j} \parallel v_{8+j} \parallel v_j)$
    $\leftarrow S_8(u_{56+i} \parallel u_{48+i} \parallel u_{40+j} \parallel u_{32+j} \parallel u_{24+j} \parallel u_{16+j} \parallel u_{8+j} \parallel u_j)$
  end
  $V_i = (v_{63}, v_{62}, \cdots, v_0)$
  $U_i \leftarrow B_P(V_i) \oplus RK_i \oplus i$
  $U_i = (u_{63}, u_{62}, \cdots, u_0)$
end
`return` $C \leftarrow U_{13}$

For MILP modeling, we describe the encryption algorithm of PIPO in a different way (Algorithm 6). Round function is described using substitution layer, permutation layer and add round key operations. Substitution layer applies 8-bit S-box (S)(Table 9) on 8 bits extracted from eight different positions of input and output bits from S-box are sent back to the same positions. Permutation layer uses a 64-bit permutation (B_P)(Table 10) on the output from S-box layer. The round keys (RK_i) and constants ($i = round\,number$) are simply XORed with the output of diffusion layer.

S_8	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	5E	F9	FC	00	3F	85	BA	5B	18	37	B2	C6	71	C3	74	9D
1	A7	94	0D	E1	CA	68	53	2E	49	62	EB	97	A4	0E	2D	D0
2	16	25	AC	48	63	D1	EA	8F	F7	40	45	B1	9E	34	1B	F2
3	B9	86	03	7F	D8	7A	DD	3C	E0	CB	52	26	15	AF	8C	69
4	C2	75	70	1C	33	99	B6	C7	04	3B	BE	5A	FD	5F	F8	81
5	93	A0	29	4D	66	D4	EF	0A	E5	CE	57	A3	90	2A	09	6C
6	22	11	88	E4	CF	6D	56	AB	7B	DC	D9	BD	82	38	07	7E
7	B5	9A	1F	F3	44	F6	41	30	4C	67	EE	12	21	8B	A8	D5
8	55	6E	E7	0B	28	92	A1	CC	2B	08	91	ED	D6	64	4F	A2
9	BC	83	06	FA	5D	FF	58	39	72	C5	C0	B4	9B	31	1E	77
A	01	3E	BB	DF	78	DA	7D	84	50	6B	E2	8E	AD	17	24	C9
B	AE	8D	14	E8	D3	61	4A	27	47	F0	F5	19	36	9C	B3	42
C	1D	32	B7	43	F4	46	F1	98	EC	D7	4E	AA	89	23	10	65
D	8A	A9	20	54	6F	CD	E6	13	DB	7C	79	05	3A	80	BF	DE
E	E9	D2	4B	2F	0C	A6	95	60	0F	2C	A5	51	6A	C8	E3	96
F	B0	9F	1A	76	C1	73	C4	35	FE	59	5C	B8	87	3D	02	FB

Table 9: 8-bit S-Box of PIPO

i	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
$B_P(i)$	0	1	2	3	4	5	6	7	15	8	9	10	11	12	13	14
i	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
$B_P(i)$	20	21	22	23	16	17	18	19	27	28	29	30	31	24	25	26
i	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47
$B_P(i)$	38	39	32	33	34	35	36	37	45	46	47	40	41	42	43	44
i	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63
$B_P(i)$	49	50	51	52	53	54	55	48	58	59	60	61	62	63	56	57

Table 10: Bit Permutation in PIPO

3.2 MILP Modeling for PIPO

The model for valid differential propagations of PIPO is constructed bitwise. In each round, subkey addition, S-box, and bit permutation operations are used. Block size in PIPO is 64-bit and it consists of 13 rounds. For 64-bit plaintext difference, binary variables $u_0, u_1, \cdots u_{63}$ represent active or inactive bits for first

$\triangle_j \rightarrow$ / $\triangle_i \downarrow$	0	1	2	3	4	5	6	7	8	9	A	B	C	D	$\cdots$	FF
0	256	0	0	0	0	0	0	0	0	0	0	0	0	0	$\cdots$	0
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	$\cdots$	0
2	0	0	0	0	0	16	0	0	0	0	0	0	0	0	$\cdots$	0
3	0	0	0	0	0	16	0	0	0	0	0	0	0	0	$\cdots$	0
4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	$\cdots$	0
5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	$\cdots$	0
6	0	0	0	0	0	16	0	0	0	0	0	0	0	0	$\cdots$	0
7	0	0	0	0	0	16	0	0	0	0	0	0	0	0	$\cdots$	0
$\cdots$															$\cdots$	
8F	0	0	0	0	0	8	0	0	0	8	0	0	0	0	$\cdots$	0
9F	0	0	16	0	0	0	0	4	2	0	0	0	0	0	$\cdots$	0
AF	0	2	16	0	0	0	0	0	0	0	0	2	0	0	$\cdots$	0
BF	0	2	0	0	0	0	0	4	0	0	0	2	0	0	$\cdots$	0
CF	0	2	0	0	0	0	0	0	2	0	0	0	0	4	$\cdots$	2
DF	0	2	16	0	0	0	0	4	0	0	4	0	0	4	$\cdots$	2
EF	0	0	16	0	0	0	0	0	2	0	0	0	2	0	$\cdots$	0
FF	0	0	0	0	0	0	0	4	0	0	2	0	0	0	$\cdots$	2

Table 11: Difference Distribution Table of PIPO

round. The variables to represent active or inactive bits in the difference after first round are updated to $u_{64}, u_{65}, \cdots u_{127}$ and so on. The variables $u_{768}, u_{769}, \cdots u_{832}$ represent the active or inactive bits in the ciphertext difference after 13 rounds. In first round, the variables representing the bits of input and output differences to S-box layer are represented as follows:

$$\begin{bmatrix} u_{56} & u_{57} & u_{58} & u_{59} & u_{60} & u_{61} & u_{62} & u_{63} \\ u_{48} & u_{49} & u_{50} & u_{51} & u_{52} & u_{53} & u_{54} & u_{55} \\ u_{40} & u_{41} & u_{42} & u_{43} & u_{44} & u_{45} & u_{46} & u_{47} \\ u_{32} & u_{33} & u_{34} & u_{35} & u_{36} & u_{37} & u_{38} & u_{39} \\ u_{24} & u_{25} & u_{26} & u_{27} & u_{28} & u_{29} & u_{30} & u_{31} \\ u_{16} & u_{17} & u_{18} & u_{19} & u_{20} & u_{21} & u_{22} & u_{23} \\ u_{8} & u_{9} & u_{10} & u_{11} & u_{12} & u_{13} & u_{14} & u_{15} \\ u_{0} & u_{1} & u_{2} & u_{3} & u_{4} & u_{5} & u_{6} & u_{7} \end{bmatrix} \rightarrow \begin{bmatrix} u_{120} & u_{121} & u_{122} & u_{123} & u_{124} & u_{125} & u_{126} & u_{127} \\ u_{113} & u_{114} & u_{115} & u_{116} & u_{117} & u_{118} & u_{119} & u_{112} \\ u_{108} & u_{109} & u_{110} & u_{111} & u_{104} & u_{105} & u_{106} & u_{107} \\ u_{101} & u_{102} & u_{103} & u_{96} & u_{97} & u_{98} & u_{99} & u_{100} \\ u_{90} & u_{91} & u_{92} & u_{93} & u_{94} & u_{95} & u_{88} & u_{89} \\ u_{83} & u_{84} & u_{85} & u_{86} & u_{87} & u_{80} & u_{81} & u_{82} \\ u_{79} & u_{72} & u_{73} & u_{74} & u_{75} & u_{76} & u_{77} & u_{78} \\ u_{70} & u_{71} & u_{64} & u_{65} & u_{66} & u_{67} & u_{68} & u_{69} \end{bmatrix}$$

The permutation layer is applied on the output from S-box layer and output of the permutation layer which acts as an input to the second round is represented as follows:

$$\begin{bmatrix} u_{120} & u_{121} & u_{122} & u_{123} & u_{124} & u_{125} & u_{126} & u_{127} \\ u_{112} & u_{113} & u_{114} & u_{115} & u_{116} & u_{117} & u_{118} & u_{119} \\ u_{104} & u_{105} & u_{106} & u_{107} & u_{108} & u_{109} & u_{110} & u_{111} \\ u_{96} & u_{97} & u_{98} & u_{99} & u_{100} & u_{101} & u_{102} & u_{103} \\ u_{88} & u_{89} & u_{90} & u_{91} & u_{92} & u_{93} & u_{94} & u_{95} \\ u_{80} & u_{81} & u_{82} & u_{83} & u_{84} & u_{85} & u_{86} & u_{87} \\ u_{72} & u_{73} & u_{74} & u_{75} & u_{76} & u_{77} & u_{78} & u_{79} \\ u_{64} & u_{65} & u_{66} & u_{67} & u_{68} & u_{69} & u_{70} & u_{71} \end{bmatrix}$$

We describe all possible propagation patterns for S-box with a system of linear inequalities.

$$e.g.(u_0, u_1, u_2, u_3, u_4, u_5, u_6, u_7 \rightarrow u_{64}, u_{65}, u_{66}, u_{67}, u_{68}, u_{69}, u_{70}, u_{71})$$

The variables corresponding to bits having the difference takes '1' and it takes '0' otherwise. A constraint $u_0 + u_1 + \cdots + u_{63} \geq 1$ is added to ensure that plaintext difference has at least one active bit.

3.2.1 DDT of 8-bit S-box. To model the 8-bit S-box of PIPO, we generate the DDT (Table 11) for each possible input and output difference (Δ_i, Δ_j) using our tool. MILES uses Algorithm 1 to generate the DDT. The entries (i, j) in the Table 11 corresponds to the number of occurrences for output differences Δ_j when the input differences were set as Δ_i. We get a 256x256 DDT for an 8-bit S-box. The non-zero values in the DDT corresponds to a possible difference propagation and zero values indicates an impossible propagation.

3.2.2 Linear Inequalities for Outer Module of MILP Model. The DDT generated in previous step is used in MILES to derive the truth table ($\star$-TT) using Algorithm 2. The $\star$-TT of PIPO contains 20621 entries which are further minimized by our tool. MILES interfaces with Espresso in 'epos' mode and the $\star$-TT is minimized to $\star$-TT$_{min}$ with 4701 entries. Using Algorithm 5, we convert each entry of $\star$-TT$_{min}$ into a linear inequality. We represent each entry of $\star$-TT$_{min}$ using 16 binary variables $(x_0, x_1, x_2, x_3, x_4, x_5, x_6, x_7, y_0, y_1, y_2, y_3, y_4, y_5, y_6, y_7)$, where first eight variables $(x_0, x_1, x_2, x_3, x_4, x_5, x_6, x_7)$ represent the input difference and remaining variables $(y_0, y_1, y_2, y_3, y_4, y_5, y_6, y_7)$ represent the output difference. These linear inequalities are used as constraints in the outer module[2] and minimization of number of active S-boxes is used as objective function.

3.2.3 Linear Inequalities for Inner Module of MILP Model. Differential probability of S-box was introduced to get MILP-model by Sun et al. in [23] and this technique was also used by Zhu et al. to present the MILP based differential attack on round-reduced GIFT in [31]. We optimize the probability of differential characteristics in the inner-MILP module. For this purpose, we need the linear inequalities for all non-zero entries in the DDT which corresponds to the

[2] https://github.com/tarunyadav/PIPO-MILP

possible difference propagation and their probabilities. In the DDT of PIPO S-box, there are seven different values for the probability of possible difference propagations i.e. 2^{-0}, $2^{-4.00}$, $2^{-4.41}$, $2^{-4.67}$, $2^{-5.00}$, $2^{-5.41}$, $2^{-6.00}$, $2^{-7.00}$ (Table 12). This requires seven extra binary variables to represent the probability of each possible propagation. MILES uses DDT to generate truth table (f-TT) with 20621 entries using Algorithm 4. Each entry of the f-TT is represented by 23 binary variables where 16 input variables $(x_0, x_1, x_2, x_3, x_4, x_5, x_6, x_7, y_0, y_1, y_2, y_3, y_4, y_5, y_6, y_7)$ represents the input and output differences. The remaining seven input variables $(p_0, p_1, p_2, p_3, p_4, p_5, p_6)$ represent the probabilities of corresponding difference propagations. MILES minimizes the f-TT to f-TT$_{min}$ with 'epos' mode of Espresso which results in 6035 entries in f-TT$_{min}$. Each entry of f-TT$_{min}$ is converted into the linear inequality using Algorithm 5. This set of linear inequalities is used to optimize the probability of differential characteristics in the block cipher PIPO. The objective function for inner module[3] is defined as minimization of equation 2 over active S-boxes (AS).

$$\sum_{\forall AS} \sum_{i=0}^{6} -\log_2(Pr_i) \times (p_0 + p_1 + p_2 + p_3 + p_4 + p_5 + p_6) \tag{2}$$

$Pr[(x_0, x_1, \cdots, x_7) \rightarrow (x_8, x_9, \cdots, x_{15})]$	$(p_0, p_1, \cdots, p_6)$
$1 = 2^{-0}$	(0,0,0,0,0,0,0)
$2/256 = 2^{-7.00}(Pr_6)$	(0,0,0,0,0,0,1)
$4/256 = 2^{-6.00}(Pr_5)$	(0,0,0,0,0,1,0)
$6/256 = 2^{-5.41}(Pr_4)$	(0,0,0,0,1,0,0)
$8/256 = 2^{-5.00}(Pr_3)$	(0,0,0,1,0,0,0)
$10/256 = 2^{-4.67}(Pr_2)$	(0,0,1,0,0,0,0)
$12/256 = 2^{-4.41}(Pr_1)$	(0,1,0,0,0,0,0)
$16/256 = 2^{-4.00}(Pr_0)$	(1,0,0,0,0,0,0)

Table 12: Binary variables to encode the Probability

3.3 Optimization of Differential Probability using MILES

We solve the MILP model using Gurobi solver [13] to optimize the probability of differential characteristics for PIPO. In the outer-MILP module, the objective function is to minimize the number of active S-boxes in the differential characteristics. We get 13 active S-boxes for 7 rounds differential characteristics in PIPO. The objective function for the inner-MILP module is to maximize the probability of differential characteristics using the positions of active S-boxes obtained in the outer module. We constructed many differential characteristics for PIPO reduced

[3] https://github.com/tarunyadav/PIPO-MILP

to 6/7 rounds. There does not exist any 6-round differential characteristic with the probability better than $2^{-54.4}$ and best differential characteristics for 7-round PIPO exists with the probability of 2^{-65}. We constructed the 7-round differential characteristics for PIPO using the inequalities generated with MILES which is shown in Table 13.

Round (i)	Input Difference (Δ_i)	Probability
0	0x0101000101000001	1
1	0x0000000000008000	2^{-4}
2	0x0000000000080080	2^{-4}
3	0x2011112000800080	2^{-11}
4	0x404100408101c080	2^{-19}
5	0x0000101000100000	2^{-16}
6	0x0000000080000000	2^{-7}
7	0x0001000004084000	2^{-4}

Table 13: 7-round Differential Characteristics for PIPO

3.4 Impossible Differential Cryptanalysis of PIPO-64/128

Impossible differential attack is opposite to differential attack. The basic idea is to use zero probability differential characteristics in place of a high probability characteristic to filter out the wrong keys [4]. For this purpose, the zero probability characteristics are constructed by proving a contradiction between the two differential characteristics of probability one each. This approach is known as miss-in-the-middle technique to search an impossible differential characteristic. Nowadays, the MILP based technique is used to search these zero probability differential characteristics. The MILP model to search the high probability differential characteristics with some added constraint is used to search the impossible differential characteristic.

To search the impossible differential, we iterate all (Δ_i, Δ_o) pairs with one active bit in the input and output. For this purpose, additional constraints to fix the input and output differences are added in the MILP model. The gurobi solver is used to solve the outer module of MILP model as discussed in section 3.2.2. The input and output differences corresponding to infeasible solution are considered as impossible differential characteristic. Using this method, we obtain the following 4-round impossible differential characteristics (Δ_0, Δ_4). However, our bound for impossible differential attack is similar to that of the designers claim.

0000000000000000000000000000000000000000000000000000000000000001→
0000000000000000000000000000000000000000000000000000000000010000

3.5 Related-key Differential Distinguisher for PIPO-64/128

Resistance to related-key attacks was not considered by the designers of PIPO-64/128 and any security claim in the related-key setting is not provided. In differential attack, the adversary is allowed to choose a difference in the plaintexts and observe the differences in ciphertexts. In related-key differential attack, the adversary is allowed to choose a relation (difference) in the key together a relation (difference) in the plaintexts [3, 18]. The adversary is allowed to get the encryption of first plaintext using the secret key and a key related to this key is used to encrypt the another plaintext. We model an MILP problem to search the related-key differential characteristic in PIPO-64/128.

3.5.1 MILP Model for Related-key Differential Characteristic. The secret key K is divided into the two 64-bit keys K_0 and K_1 which are used as round subkeys in PIPO-64/128. We model the similar MILP problem to search the related-key differential characteristic as described in section 3.2. Additionally, we need to model the key addition layer and solve the MILP model in order to get the optimal related-key characteristics in PIPO-64/128 [23].

Modeling Key Addition Layer. We need to introduce the additional constraints in the MILP model corresponding to the round keys. The 128 new variables are introduced corresponding to the 128-bit secret key. The 64 key variables are added in one round and the other 64 key variables are added in the subsequent round. To add the constraints for key addition layer, for each bit of input x_i and key k, we follow the conditions on bit variables to exclude the impossible patterns (Equation 3). Here, x_i and k refer to the input bit and corresponding key bit. The bit variable x_{i+1} is an output of the XOR operation *i.e.* $x_{i+1} = k \oplus x_i$.

$$\begin{aligned} x_i + k - x_{i+1} &\geq 0 \\ x_i - k + x_{i+1} &\geq 0 \\ -x_i + k + x_{i+1} &\geq 0 \\ x_i + k + x_{i+1} &\leq 2 \end{aligned} \tag{3}$$

3.5.2 Full-round Related-key Differential Distinguisher for PIPO-64/128. We solve the MILP model to search the related-key differential characteristics using gurobi solver. We get a 2-round iterative related-key characteristic with the probability of 2^{-4}. The optimal related-key differential characteristic for full round PIPO-64/128 is obtained with a probability of 2^{-24} using 2-round iterative characteristic (Table 14). We also get full-round characteristics with probability of 2^{-28} under zero difference in the plaintext as well as in the ciphertext (Table 15).

3.5.3 Collisions in PIPO-64/128. The zero difference related-key characteristics will lead to a collision in the hash function designed using PIPO-64/128. We searched for the existence of input and output pairs under different keys following zero difference characteristic (collision). We encrypt the 2^{28} random samples under related keys and one such pair is expected in each experiment. Therefore, we can construct as many samples providing us the collision in the input and output under the different keys. We have verified these plaintext and ciphertext samples by using the designers program. One such collision in PIPO-64/128 is presented in the Table 15.

Round (r)	Difference (Δ_r) $\Delta K=$ 0x00200000020000000000040000801001000	Probability
0	0x0040000801001000	1
1	0x0020000020000000	1
2	0x0000000000000000	2^{-4}
3	0x0020000020000000	1
4	0x0000000000000000	2^{-4}
5	0x0020000020000000	1
6	0x0000000000000000	2^{-4}
7	0x0020000020000000	1
8	0x0000000000000000	2^{-4}
9	0x0020000020000000	1
10	0x0000000000000000	2^{-4}
11	0x0020000020000000	1
12	0x0000000000000000	2^{-4}
13	0x0020000020000000	1

Table 14: 13-round Characteristics with Probability 2^{-24}

4 Conclusion

In this paper, we have presented a new tool (MILES) that minimizes the linear inequalities corresponding to full difference distribution table of large S-boxes. Using MILES, there will be no need to split the DDT into p-DDT and full DDT can be used to optimize the probability of differential characteristics. The linear inequalities for AES, SKINNY and PIPO-64/128 S-boxes are constructed and results are compared with p-DDT approach. We have demonstrated the use of MILES to optimize the probability of differential characteristics in PIPO-64/128. We use the linear inequalities generated using MILES to model the MILP problem for searching the impossible and related-key differential characteristics. We have presented the full round related-key differential distinguisher and presented collision in the plaintext and ciphertext using different keys.

Round (r)	$E_K(P_r)$, $K = (K_1 \| K_0)$ K_1= 0x6DC416DD779428D2 K_0= 0x7E1D20AD2E152297	$E_{K'}(P_r^{'})$, $K^{'} = (K_1^{'} \| K_0^{'})$ $K_1^{'} = K_1 \oplus$ 0x0040000801001000 $K_0^{'} = K_0 \oplus$ 0x0020000020000000	Difference ($\Delta_r = P_r \oplus P_r^{'}$)	Probability
0	0xFFEAF697D7FCE742	0xFFEAF697D7FCE742	0x0000000000000000	1
1	0xD76EFD65756940C0	0xD76EFD65756940C0	0x0000000000000000	2^{-4}
2	0x4FA59C5858EDC4FF	0x4F859C5878EDC4FF	0x0020000020000000	1
3	0x8F6ACEC7A220C121	0x8F6ACEC7A220C121	0x0000000000000000	2^{-4}
4	0x406AD151D57A997B	0x404AD151F57A997B	0x0020000020000000	1
5	0xC5F53C44C408AC2D	0xC5F53C44C408AC2D	0x0000000000000000	2^{-4}
6	0xCFD8867C58BFCFD9	0xCFF8867C78BFCFD9	0x0020000020000000	1
7	0xC99B445F8E203697	0xC99B445F8E203697	0x0000000000000000	2^{-4}
8	0xD12CCC87E5585504	0xD10CCC87C5585504	0x0020000020000000	1
9	0x01D75CDC373A6F41	0x01D75CDC373A6F41	0x0000000000000000	2^{-4}
10	0x41C1CE1756D7C045	0x41E1CE1776D7C045	0x0020000020000000	1
11	0x388794675E6B5EDE	0x388794675E6B5EDE	0x0000000000000000	2^{-4}
12	0x1FDB4194BF26AC3B	0x1FFB41949F26AC3B	0x0020000020000000	1
13	0xCDE57DF09ECF4F7D	0xCDE57DF09ECF4F7D	0x0000000000000000	2^{-4}

Table 15: Zero Difference Characteristics with an Example of Collision

References

1. Abdelkhalek, A., Sasaki, Y., Todo, Y., Tolba, M., Youssef, A.M.: MILP modeling for (large) Sboxes to optimize probability of differential characteristics. IACR Transactions on Symmetric Cryptology, ISSN 2519-173X, vol. 2017, No. 4, pp. 99–129, DOI:10.13154/tosc.v2017.i4.99-129 (2017)
2. Bogdanov, A., Knudsen, L.R., Leander, G., Paar, C., Poschmann, A., Robshaw, M. J. B., Seurin, Y., and Vikkelsoe,C.: PRESENT: An Ultra-Lightweight Block Cipher. In Pascal Paillier and Ingrid Verbauwhede, editors, CHES 2007, vol. 4727 of LNCS, pp. 450–466, Springer (2007)
3. Biham, E.: New types of cryptanalytic attacks using related keys, Journal of Cryptology, 7(4), pp. 229-246 (1994)
4. Biham, E., Biryukov, A., and Shamir, A.: Cryptanalysis of Skipjack Reduced to 31 Rounds using Impossible Differentials, In J. Stern, editor, Advances in Cryptology, EUROCRYPT'99, LNCS, vol. 1592, pp. 12-23. Springer Verlag, 1999.
5. Bogdanov, A.: Analysis and Design of Block Cipher Constructions. Ph.D. thesis (2009)
6. Biham, E., Shamir, A.: Differential cryptanalysis of DES-like Cryptosystems. Journal of Cryptology, vol. 4, pp. 3–72, springer (1991)
7. Biham, E., Shamir, A.: Differential Cryptanalysis of the full 16-round DES. CRYPTO 92, LNCS, vol. 740, pp. 487-496, Springer (1992)
8. 'CryptoMiniSat5', https://www.msoos.org/cryptominisat5
9. IBM ILOG. IBM ILOG CPLEX Optimization Studio V12.7.0 documentation. Official webpage, https://www-01.ibm.com/software/websphere/products/optimization/cplex-studio-community-edition/ (2016)
10. Espresso Logic Minimizer, https://ptolemy.berkeley.edu/projects/embedded/pubs/downloads/espresso/
11. Gerault, D., Lafourcade, P., Minier, M., *et al.*: Revisiting AES related-key differential attacks with constraint programming. Cryptology ePrint Archive (2017)
12. Gohr, A.: Improving attacks on round-reduced SPECK32/64 using deep learning. In Boldyreva, A., Micciancio, D., editors, Advances in Cryptology- CRYPTO 2019, Cham, Springer International Publishing, pp. 150-179 (2019)

13. Gurobi Optimizer 7.5.2', http://www.gurobi.com
14. Hays, H.M.: A Tutorial on Linear and Differential Cryptanalysis. Cryptologia, vol. 26, No. 3, pp. 188-221 (2002)
15. Kim, H., Jeon, Y., Kim, G., Kim, J., Sim, B., Han, D., Seo, H., Kim, S., Hong, S., Sung, J., Hong, D.: PIPO: A Lightweight Block Cipher with Efficient Higher-Order Masking Software Implementations. In D. Hong, editors, ICISC 2020, LNCS, vol. 12593, pp. 99–122 (2021)
16. Knudsen, L., Robshaw, M.J.B.: Block Cipher Companion. Book Springer, ISBN 978-3-642-17341-7 (2011)
17. Kumar, M., Suresh, TS, Pal, S.K., Panigrahi, A.: Optimal Differential Trails in Lightweight Block Ciphers ANU and PICO. Cryptologia, vol. 44, No. 1, pp. 68-78 (2020)
18. Kelsey, J., Schneier, B. and Wagner, D.: Related-Key Cryptanalysis of 3-WAY, Biham-DES, CAST, DES-X, NewDES, RC2, and TEA, ICICS '97 Proceedings, Springer-Verlag, November 1997, pp. 233-246 (1997)
19. 'Logic friday', http://sontrak.com/
20. Matsui, M.: On correlation between the order of S-boxes and the strength of DES. Proceeding of International Conference. EUROCRYPT, Italy, May 1994, pp. 366–375 (1994)
21. Mouha, N., Wang, Q., Gu, D. and Preneel, B.: Differential and Linear Cryptanalysis Using Mixed-Integer Linear Programming. In Chuankun Wu, Moti Yung, and Dongdai Lin, editors, Inscrypt 2011, vol. 7537, LNCS, pp. 57–76. Springer (2011)
22. National Institute of Standards and Technology. Lightweight Cryptography, Finalists, https://csrc.nist.gov/projects/lightweight-cryptography/finalists. NIST (2021)
23. Sun, S., Hu, L., Wang, P., Qiao, K., Ma, X., and Song, L.: Automatic Security Evaluation and (Related-key) Differential Characteristic Search: Application to SIMON, PRESENT, LBlock, DES(L) and Other Bit-Oriented Block Ciphers. In Palash Sarkar and Tetsu Iwata, editors, ASIACRYPT 2014, Part I, vol. 8873, LNCS, pp. 158–178, Springer (2014)
24. Sun, S., Hu, L., Wang, M., Wang, P., Qiao, K., Ma, X., Shi, D., Song, L., and Fu,K.: Towards Finding the Best Characteristics of Some Bit-oriented Block Ciphers and Automatic Enumeration of (Related-key) Differential and Linear Characteristics with Predefined Properties. Cryptology ePrint Archive, Report 2014/747 (2014)
25. 'SAGE', http://www.sagemath.org/index.html
26. Sasaki, Y., and Todo, Y.: New Differential Bounds and Division Property of Lilliput: Block Cipher with Extended Generalized Feistel Network. In Roberto Avanzi and Howard Heys, editors, SAC 2016, vol. 10532 of LNCS, pp. 264–283. Springer (2016)
27. Sasaki, Y., and Todo, Y.: New Impossible Differential Search Tool from Design and Cryptanalysis Aspects - Revealing Structural Properties of Several Ciphers. In Jean-Sébastien Coron and Jesper Buus Nielsen, editors, Advances in Cryptology - EUROCRYPT 2017 - 36th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Paris, France, April 30 - May 4, 2017, Proceedings, Part III, vol. 10212 of Lecture Notes in Computer Science, pp. 185–215, Springer (2017)
28. Sasaki, Y., and Todo, Y.: New Algorithm for Modeling S-box in MILP Based Differential and Division Trail Search. In Emil Simion and Pooya Farshim, editors, SecITC 2017, vol. 10543 of LNCS, pp. 150–165. Springer (2017)
29. Sun, S., Gerault, D., Lafourcade, P., Yang, Q., Todo, Y., Qiao, K., and Hu, L.: Analysis of AES, SKINNY, and Others with Constraint Programming. IACR Trans. Symmetric Cryptol., 2017(1), pp. 281–306 (2017)

30. Yadav, T., Kumar, M.: PIPO-64/128 Distinguisher: Machine Learning based Generic Extension for Differential Cryptanalysis. In Longa, P., Rafols, C., editors, Progress in Cryptology- LATINCRYPT 2021, 7^{th} International Conference on Cryptology and Information Security in Latin America, Bogota, Colombia, October 6-8, 2021, vol. 12912 (2021)
31. Zhu, B., Dong, X., Yu, H.: MILP-Based Differential Attack on Round-Reduced GIFT. In: M. Matsui, editor, Topics in Cryptology - CT-RSA 2019, San Francisco, CA, USA, pp. 372-390 (2019)